

# 阿里云 数据库文件存储

## SDK参考

文档版本：20200401

# 法律声明

---

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

## 通用约定

| 格式                                                                                | 说明                                 | 样例                                                                                                                       |
|-----------------------------------------------------------------------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|  | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。   |  <b>禁止：</b><br>重置操作将丢失用户配置数据。          |
|  | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  <b>警告：</b><br>重启操作将导致业务中断，恢复业务时间约十分钟。 |
|  | 用于警示信息、补充说明等，是用户必须了解的内容。           |  <b>注意：</b><br>权重设置为0，该服务器不会再接受新请求。    |
|  | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。       |  <b>说明：</b><br>您也可以通过按Ctrl + A选中全部文件。  |
| >                                                                                 | 多级菜单递进。                            | 单击设置 > 网络 > 设置网络类型。                                                                                                      |
| <b>粗体</b>                                                                         | 表示按键、菜单、页面名称等UI元素。                 | 在结果确认页面，单击确定。                                                                                                            |
| Courier字体                                                                         | 命令。                                | 执行cd /d C:/window命令，进入Windows系统文件夹。                                                                                      |
| ##                                                                                | 表示参数、变量。                           | bae log list --instanceid<br>Instance_ID                                                                                 |
| [ ]或者[a b]                                                                        | 表示可选项，至多选择一个。                      | ipconfig [-all -t]                                                                                                       |
| { }或者{a b}                                                                        | 表示必选项，至多选择一个。                      | switch {active stand}                                                                                                    |

# 目录

---

|                   |   |
|-------------------|---|
| 法律声明.....         | I |
| 通用约定.....         | I |
| 1 c/c++头文件说明..... | 1 |

# 1 c/c++头文件说明

本文档主要为您说明在通过SDK方式接入DBFS服务时所需要遵循的接口使用规范。

DBFS的SDK头文件位于“/usr/local/include/dbfs.h”。以下为DBFS服务目前支持的接口，具体语义可参考Linux MAN手册。

```
int dbfs_mount(const char *conf_file);
int dbfs_umount();
int dbfs_open(const char *pathname, int flags, ...);
int dbfs_close(int fd);
int dbfs_fcntl(int fd, int cmd, ...);
off_t dbfs_lseek(int fd, off_t offset, int whence);
int dbfs_fallocate(int fd, int mode, off_t offset, off_t len);
int dbfs_posix_fallocate(int fd, off_t offset, off_t len);
int dbfs_truncate(const char *path, off_t length);
int dbfs_ftruncate(int fd, off_t length);
int dbfs_fsync(int fd);
int dbfs_fdatasync(int fildes);
int dbfs_sync_file_range(int fd, off64_t offset, off64_t nbytes,
unsigned int flags);
int dbfs_rename(const char *oldname, const char *newname);
int dbfs_link(const char *oldpath, const char *newpath);
int dbfs_unlink(const char *filepath);
int dbfs_symlink(const char *oldpath, const char *newpath);
ssize_t dbfs_readlink(const char *path, char *buf, size_t bufsiz);
int dbfs_lstat(const char *path, struct stat *buf);
int dbfs_stat(const char *path, struct stat *buf);
int dbfs_fstat(int fd, struct stat *buf);
int dbfs_statvfs(const char *path, struct statvfs *buf);
int dbfs_chdir(const char *path);
char *dbfs_getcwd(char *buf, size_t size);
int dbfs_mkdir(const char *dirname, mode_t mode);
int dbfs_rmdir(const char *dirname);
int dbfs_access(const char *path, int mode);
int dbfs_flock(int fd, int operation);
int dbfs_mkstemp(char *tmpl);
int dbfs_dup2(int oldfd, int newfd);
int dbfs_chown(const char *path, uid_t owner, gid_t group);
int dbfs_chmod(const char *path, mode_t mode);
int dbfs_fchmod(int fd, mode_t mode);
char *dbfs_realpath(const char *path, char *resolved_path);
ssize_t dbfs_read(int fd, void *buf, size_t size);
ssize_t dbfs_write(int fd, const void *buf, size_t size);
ssize_t dbfs_pread(int fd, void *buf, size_t size, off_t offset);
ssize_t dbfs_pwrite(int fd, const void *buf, size_t size, off_t
offset);
DIR *dbfs_opendir(const char *name);
int dbfs_closedir(DIR *dirp);
struct dirent *dbfs_readdir(DIR *dirp);
int dbfs_readdir_r(DIR *dirp, struct dirent *entry, struct dirent **
result);
FILE *dbfs_fopen(const char *path, const char *mode);
FILE *dbfs_fdopen(int fildes, const char *mode);
size_t dbfs_fread(void *ptr, size_t size, size_t nmemb, FILE *stream
);
size_t dbfs_fwrite(const void *ptr, size_t size, size_t nmemb, FILE
*stream);
int dbfs_fflush(FILE *stream);
```

```
int dbfs_fclose(FILE *stream);
int dbfs_fileno(FILE *stream);
int dbfs_feof(FILE *stream);
int dbfs_ferror(FILE *stream);
void dbfs_clearerr(FILE *stream);
int dbfs_fseeko(FILE *stream, off_t offset, int whence);
int dbfs_fseek(FILE *stream, long offset, int whence);
off_t dbfs_ftello(FILE *stream);
long dbfs_ftell(FILE *stream);
void dbfs_rewind(FILE *stream);
int dbfs_fputs(const char *s, FILE *stream);
int dbfs_fputc(int c, FILE *stream);
int dbfs_putc(int c, FILE *stream);
char *dbfs_fgets(char *s, int size, FILE *stream);
int dbfs_fgetc(FILE *stream);
int dbfs_getc(FILE *stream);
int dbfs_fprintf(FILE *stream, const char *format, ...);
int dbfs_fscanf(FILE *stream, const char *format, ...);
int dbfs_vfscanf(FILE *stream, const char *format, va_list args);
// POSIX async io interface
void dbfs_io_prep_pread(struct iocb *iocb, int fd, void *buf, size_t
count, long long offset);
void dbfs_io_prep_pwrite(struct iocb *iocb, int fd, void *buf,
size_t count, long long offset);
int dbfs_io_setup(int maxevents, io_context_t *ctxp);
int dbfs_io_destroy(io_context_t ctx);
int dbfs_io_submit(io_context_t ctx, long nr, struct iocb *ios[]);
int dbfs_io_cancel(io_context_t ctx, struct iocb *iocb, struct
io_event *evt);
int dbfs_io_getevents(io_context_t ctx_id, long min_nr, long nr,
struct io_event *events, struct timespec *timeout);
```